FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

ASSIGNMENT OF BACHELOR'S THESIS

Title:	Avast Cybersecurity Phishing Game
Student:	Eder Jair Tejada Ortigoza
Supervisor:	Richard Barry, Ph.D.
Study Programme:	Informatics
Study Branch:	Web and Software Engineering
Department:	Department of Software Engineering
Validity:	Until the end of summer semester 2020/21

Instructions

Avast Software s.r.o. Threat Labs team, want to showcase how easy it is for a web user to fall into the trap of phishing scams sites. The objective of the project is to build a gamified web application used for educational purposes at international events such as Cybersecurity and AI events.

Requirements: The game needs to be web-based and accessible online. The first iteration of the game will deal with images and URLs contained in the data source (Provided by the Threat Labs team). Second, it will add the feature of highlighting actual errors on the fake sites. The application should handle user input and provide a score based on the correct answers.

Project Life cycle: Collaborate with UX/UI specialists and visual designers to ensure visual and functional requirements. Develop the application in HTML, CSS, JS and NodeJS. Set up a publicly accessible server that could showcase the application. Do user testing and make improvements before events.

References

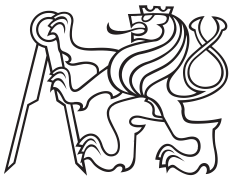
Will be provided by the supervisor.

Ing. Michal Valenta, Ph.D.
Head of Department

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
Dean

Prague October 21, 2019

Bachelor Project



Czech Technical University in Prague

F8

Faculty of Information Technology
Department of Software Engineering

Avast Cybersecurity Phishing Game

Eder Jair Tejada Ortigoza

Supervisor: Richard Barry, Ph.D.

Prague, December 2019

Acknowledgements

Foremost, I want to thank my parents for their love and unconditional support during all my studies. Also, to my siblings, Marco and Viridiana, for being my inspiration and give me advice whenever was needed. Despite the odds, my family never held me back and always push me to go further. Everything that I accomplished is because of their great effort and countless sacrifices.

Many thanks to my host parents Małgorzata and Ludvík for their incredible help during all these years. They welcomed me to their family and made possible my studies here. I am fortunate to have them in my life, and I will never forget their kindness.

I want to extend my gratitude to my supervisor, Richard Barry, for giving me the chance to work on this project and all his assistance. To Monika Poštová for the coordination of all the project. To Filip Kudrev and Michaela Nováková for their help during the UI and UX design development. To Miroslaw Ratman for his support during the backend development. To Tomáš Trnka for sharing his knowledge on this field and provide me with the data and tools needed to create the game. To my team, *Go Local*, for their motivation and valuable feedback. And to everyone else in Avast that contributed to this project.

Finally, I want to express my most profound appreciation to the others that support me on this journey. To our family friends, Omar and Luz, for helping me to fulfil my studies. To the Hamr family (Bětka, Jan, Daniela, Mariana and Adam) for all the good times that we shared together. And to my aunt Patricia, although she is no longer with us, I want to dedicate this work to her memory.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended. In accordance with Article 46(6) of the Act, I hereby grant a nonexclusive authorization (license) to utilize this thesis, including any and all computer programs incorporated therein or attached thereto and all corresponding documentation (hereinafter collectively referred to as the "Work"), to any and all persons that wish to utilize the Work. Such persons are entitled to use the Work in any way (including for-profit purposes) that does not detract from its value. This authorization is not limited in terms of time, location and quantity. However, all persons that makes use of the above license shall be obliged to grant a license at least in the same scope as defined above with respect to each and every work that is created (wholly or in part) based on the Work, by modifying the Work, by combining the Work with another work, by including the Work in a collection of works or by adapting the Work (including translation), and at the same time make available the source code of such work at least in a way and scope that are comparable to the way and scope in which the source code of the Work is made available.

In December 10, 2019 . . . . . .

Citation of this thesis

Eder Jair Tejada Ortigoza. Avast Cybersecurity Phishing Game. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2019.

Abstract

Nowadays, phishing attacks are on the rise and more sophisticated than ever since they are incredibly profitable for cybercriminals. These attacks consist of social engineering techniques via digital channels such as web, email, social media, mobile applications and any other form of digital communication. One of the main reasons people become victims of phishing is because they are not able to detect fake from legitimate sites.

To solve this problem, the outcome of this thesis is a gamified web application to educate people around the world to identify phishing sites. Similarly, to create awareness of this topic and use it at international events, sponsored by Avast, or other external channels like blogs and social media.

After releasing the game application to the public, we found that only about 60-70 % of users tested could identify the scam sites correctly. Although this number seems positive, it is scarily low based on the fact that we showed only one type of phishing (i.e. scam websites). We estimate that running a test, on a similar user sample, but with other types of phishing leads to a meaningful impact on negative results.

From a security perspective, this study emphasizes the need to aid in protecting potential victims against phishing, and the significance of this project for the upcoming years.

Keywords: phishing game, cybersecurity, gamified web application, social engineering

Supervisor: Richard Barry, Ph.D.

Abstrakt

V dnešní době jsou útoky phishingu na vzestupu a sofistikovanější než kdy předtím, protože jsou pro počítačové zločince neuvěřitelně výhodné. Tyto útoky spočívají v technikách sociálního inženýrství prostřednictvím digitálních kanálů, jako je web, e-mail, sociální média, mobilní aplikace a jakákoli jiná forma digitální komunikace. Jedním z hlavních důvodů, proč se lidé stávají oběťmi phishingu, je skutečnost, že nejsou schopni odhalit skutečné z falešných stránek.

Abychom tento problém vyřešili, je výsledkem této práce gamifikovaná webová aplikace, která vzdělává lidi po celém světě k identifikaci phishingových webů. Stejně tak je třeba zvýšit povědomí o tomto tématu a použít jej na mezinárodních akcích sponzorovaných společností Avast nebo na jiných externích kanálech, jako jsou blogy a sociální média.

Po vydání herní aplikace pro veřejnost jsme zjistili, že pouze asi 60-70 % testovaných uživatelů dokázalo správně identifikovat podvodné weby. Přestože se toto číslo jeví jako pozitivní, je děsivé nízké na základě skutečnosti, že jsme projevili pouze jeden typ phishingu (tj. Podvodné weby). Odhadujeme, že provedení testu na podobném vzorku uživatelů, ale s jinými typy phishingu, by měl významný dopad na negativní výsledky.

Z hlediska bezpečnosti tato studie zdůrazňuje potřebu pomoci při ochraně potenciálních obětí před phishingem a význam tohoto projektu pro nadcházející roky.

Klíčová slova: phishingová hra, kybernetická bezpečnost, gamifikovaná webová aplikace, sociální inženýrství

Contents

Figures

Code listing

Chapter 1

Introduction

Avast Software s.r.o. is a global company dedicated to providing safety and privacy for everyone and their famous antivirus protects millions of users against the most dangerous threats online (e.g. Spyware, Malware, Ransomware, Phishing). Technology is rapidly evolving, and cyber crimes are not the exception, every day people around the world become victims of cyber-attacks.

1.1 Motivation

This thesis addresses the problem of phishing, where the attacker tricks the user into revealing his personal information (i.e. passwords, bank account numbers, credits cards) [1]. To achieve this, criminals use social engineering techniques by appealing to people's vanity, greed, curiosity, altruism, or respect for or fear of authority to get them to reveal this type of sensitive information or access to an IT system [2].

According to independent research, phishing attacks targeting American institutions grew by 40% in 2018, and 80% of them targeted some of the leading industries (e.g. Financial, Email, Cloud, Payment services and SaaS). In 2018, the Federal Bureau of Investigation (FBI) reported that companies all around the world lost $12.0 billion because of compromised business email accounts [3].

Most of these attacks happen through email or fake websites but also include telephone or smishing (i.e. SMS phishing). Phishing is still the leading attack

method by cybercriminals because of its impact and scale. The main reason people are not able to detect real from phishing sites is that they look and feel like exact copies of legitimate websites.

The devastating consequences of successful phishing attacks on companies could include loss of customers due to data breach and brand trust, as well as in many cases bankruptcy to smaller and mid-sized businesses as everyone is a target, regardless of their size. Hence, it is crucial to create awareness about the topic and educate people about how to identify and avoid phishing sites.

1.2 Strategy

In order to solve this problem, we needed to develop a strategy that ensures success and fulfilment of our primary goal. The goal of this project is to educate and engage people on the topic of phishing. The way to achieve this is to base our research and final output on the following eight dimensions of quality:

1. ***Performance***: In section 2.3, we defined requirements and characteristics of the application.
2. ***Features***: In section 3.2, we describe the qualities that enhance the main functionalities of the application — moreover, the inclusion of other features to improve the overall performance, as explained in section 3.3.
3. ***Reliability***: From the analysis in chapter 2, it is undeniable the rise of phishing attacks. Hence, the reliability of the application for the upcoming years.
4. ***Conformance***: Documentation for the code is included for further development and to meet the production standards.
5. ***Durability***: Since phishing trends are continually changing it is crucial to update the content and maintain the application regularly to keep relevancy and functionality.
6. ***Serviceability***: In order to guarantee the service of the application and avoid breakdowns during events, it is necessary to do user testing beforehand.
7. ***Aesthetics***: For this project, we work together with UX and UI specialist to assure the aesthetics of the application. Design patterns were taken into consideration to respect the brand identity.

1.3 Structure of the Thesis

This thesis is organized as follows.

In Chapter 2, we analyze further the problem to solve with real examples that demonstrate the impact of phishing on the last years and its behaviour. Moreover, define the goals for the project and the assigned solution and requirements.

In Chapter 3, we describe the development process from the early stages to the actual deployment. Similarly, present some of the critical factors used to successfully built the application. Finally, identifying issues of the current implementation and how they were solved.

In Chapter 4, we demonstrate some use cases and the data obtained to support our research during the analysis. The results show the importance to emphasize this topic and create awareness.

In Chapter 5, we conclude the thesis and outline future work on this project.

Chapter 2

Analysis

The Avast Threat Lab team is continuously working on the detection and eradication of different types of cyber-attacks. Recently, their researchers helped the international police to neutralize 850,000 attacks of a malware known as *Retadup*, which distributed a malicious cryptocurrency miner and other malware to infected devices [4]. The previous case is just an example of all the threats happening every day around the world, being phishing one of the most popular ones due to its profitability.

Phishing attempts grew by 65% in 2017 [5], and nearly 1.5 million sites of this type created every month [6]. The average phishing attack costs a mid-sized company $1.6 million [5]. Most of these attacks occur by spam emails, which represents 54% of all emails sent globally [7]. Moreover, in 2017, an average user was receiving up to 16 malicious spam emails per month [8].

There are different types of phishing:

- **Phishing scams**: Target personal identity and financial information.
- **IT/SaaS phishing**: Targets access to organizations credentials and data.
- **SMS phishing**: Target mobile phones using text messages to obtain personal information.
- **Voice phishing**: Target landline telephone systems to gain access to private personal and financial information.

In 2018, email statistics from *Office 365* reported to block 5 billion phish emails

and identify 8 million business compromise attempts. The same report mentioned that 20% of users click on malicious links after 5 minutes [9]. Phishing emails are usually less successful thanks to the advances on classifying them as spam, but some of them manage to slip into inboxes.

Organizations in the United States remain the most popular for phishers in 2018, accounting 84% of total phishing volume. Concerning increment on phishing volume, Canada is one of the most outstanding with an increase of 173% (i.e. 4% of total volume). Turkey had a shocking increment of 905% but still represents only 1% of the total worldwide [3].

Often, phishing sites try to look legitimate by using HTTPS. For most of the users when a browser shows HTTPS abbreviation and a green padlock symbol, it represents that a site is secured. A survey, conducted by the *Anti-Phishing Working Group* (APWG), show that 80% of respondents believe the green padlock symbol indicates that a site is legit and safe [10], which is not the case.The APWG second-quarter report of 2019 [11] showed that at least 50% of phishing sites were using SSL.

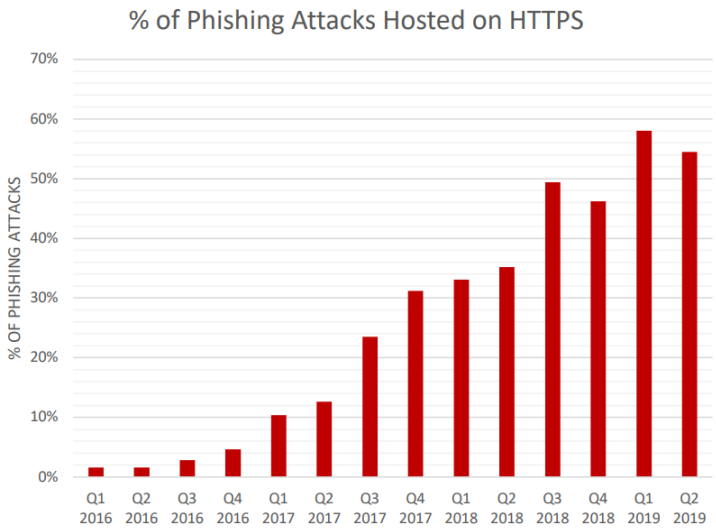


Figure 2.1: APWG graph showing phish attacks hosted on HTTPS.

Another problem is the use of free hosting providers, which has been increasing tremendously and represents 13% of phishing volume. Criminals use this hosting type because it allows them to create a large volume of sites for no cost in a short time. One of the most popular providers of this kind is *000webhostapp*, accounting 69% of free hosted phish sites [3]. Other examples are outdated *WordPress* sites that can be easily hacked and use to run phishing campaigns. Generally, the price to deploy a phishing kit is roughly $26 [12].

Moreover, cybercriminals use social engineering techniques to trick people into giving up personal information and login credentials. Despite phishing there are other types of social engineering [13], some examples are the following:

- **Baiting**: Offer something desirable as a way to download a malicious file.
- **Pretexting**: Pretend to be someone else to get access to privileged data.
- **Scareware**: Trick someone into thinking their computer is infected and offer a "solution" that contains a virus.

A real example of this is the US Department of Justice's 2016 data breach [14], where 200GB of data was exposed. The hacker used social engineering to impersonate a member of the staff and convince others to provide access to internal files. Another case is the 2016 US elections scandal with *Cambridge Analytica* [15], where personal data was used to influence public opinion. These cases demonstrate how much phishing can scale and therefore, the importance to create consciousness and educate users to prevent these attacks.

For this project, we collaborated with malware specialists from the Avast Threat Lab team to acquire the data set needed to build the application. This chapter will describe the project life cycle to achieve the desired outcome. Finally, the assigned solution will be defined together with the functional and non-functional requirements for this project.

2.1 Project Life Cycle

Previously, Martin Hron [16] (Malware Researcher at Avast) created the same application. However, the implementation was not suitable for its use and required some improvement. Therefore, some goals had to be set to ensure the successful development of the new version. These goals are as follows:

1. Determine a strategy for game functionality and effectiveness during the project briefing stage.
2. Communicate and establish a relationship with malware analysts from the Avast Threat Lab team, and to collaborate on acquiring the correct data sets needed to develop the application.

3. Work together with UX/UI specialists and visual designers, to ensure that both the visual and functional requirements are aligned and possible.

4. Set up a server accessible publicly that could showcase the application and handle the data without any direct connection to a live database for security purposes.

5. Develop the application using web technologies such as HTML, CSS, JavaScript and NodeJS.

6. Do user testing and make improvements before the actual events.

2.2 Assigned Solution

This section will describe in detail the requested solution, which is the primary output of this thesis. A gamified web application must be implemented to showcase data provided by the Avast Threat Lab team. The main objective is to educate and engage consumers on the topic and give a positive brand impression of Avast. The requirements, functional and non-functional, will be described in the upcoming section.

2.3 Functional Requirements

The elementary functionalities that the outcome of this project must accomplish are:

F1 Display a set of data containing images and URLs hosted on the Avast CDN.

F2 Provide an option to select the correct answer (i.e. real or fake site).

F3 Generate a score based on the given answers.

F4 Screen size responsiveness.

F5 Keep track of game results.

2.4 Non-functional Requirements

The way how these functionalities will work is:

NF1 Limited to 5 images per game. The data must be presented uniquely and randomly (i.e. different set of content on each try).

NF2 Answers interpreted by buttons.

NF3 Correct answers will get 100 points or zero otherwise.

NF4 Adjust website elements based on the screen size (i.e. desktop, tablet or mobile).

NF5 Keep track of game results without using a database (e.g. *Google Analytics*).

Chapter 3

Application development

This chapter will cover in detail all the development process. Starting with the User Experience (UX) and User Interface (UI) design, which were crucial to ensure the proper functionality of the game. Then, describe how the application was built and set up for deployment. Finally, analyze the given feedback and discuss possible improvements in the current implementation.

3.1 UX and UI Design

Based on the functionalities defined in the requirements, we needed to create wireframes to map the content and interaction of the application. This wireframe allowed to see each stage of the game and determine the behaviour of every component. Desktop and mobile versions consist of the next screens and elements:

1. Introduction
    - Header
    - Brief introduction
    - ***Actions***: Start
2. Game Play
    - Indicator of the game position
    - URL + Image

- Current Score
- ***Actions***: Fake, Real and Reset
- ***States***: Correct or Incorrect

3. Summary

- Overall score
- Evaluation according to score
- ***Actions***: Download and Reset
- ***States***: Perfect, Ok or Disaster

Figure 3.1: Game flow and screens.

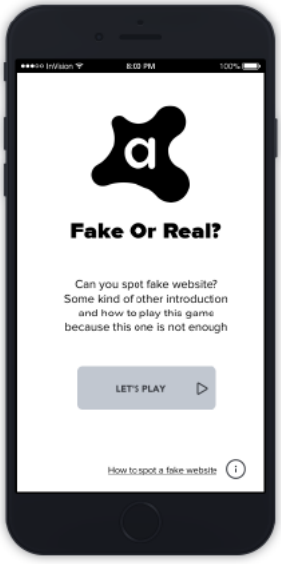


Figure 3.2: Introduction screen for mobile.

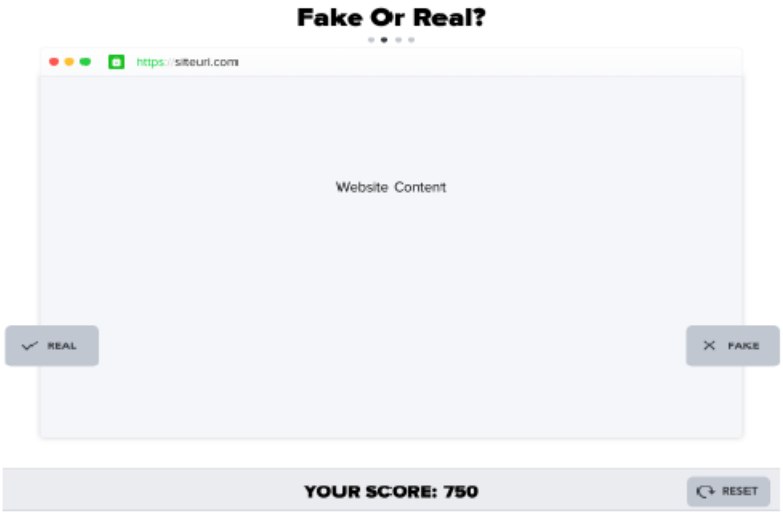


Figure 3.3: Game screen for desktop.

After each stage of the application was defined, the next step was to develop the UI design. This design had to respect different brand elements such as typography and palette. Similarly, it had to be intuitive for users and illustrate a playful mood to make the topic more interesting and entertaining.

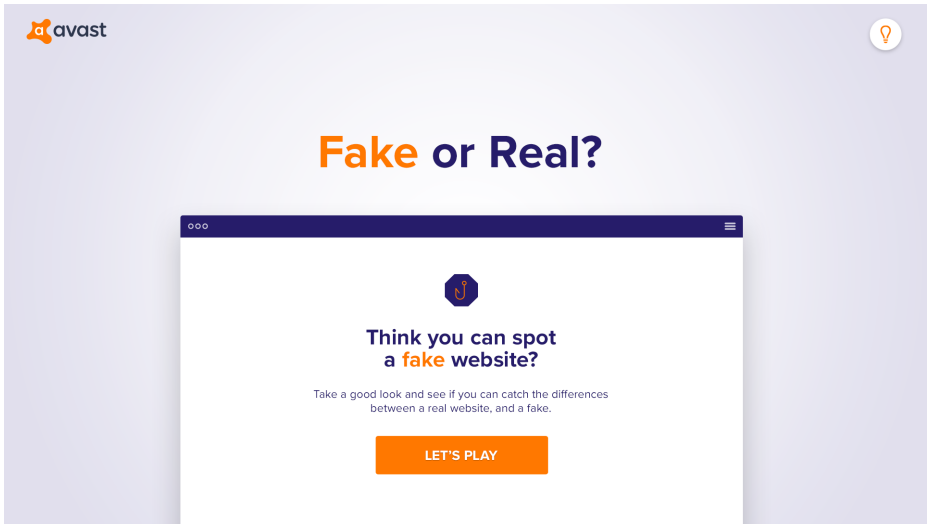


Figure 3.4: Introduction screen.

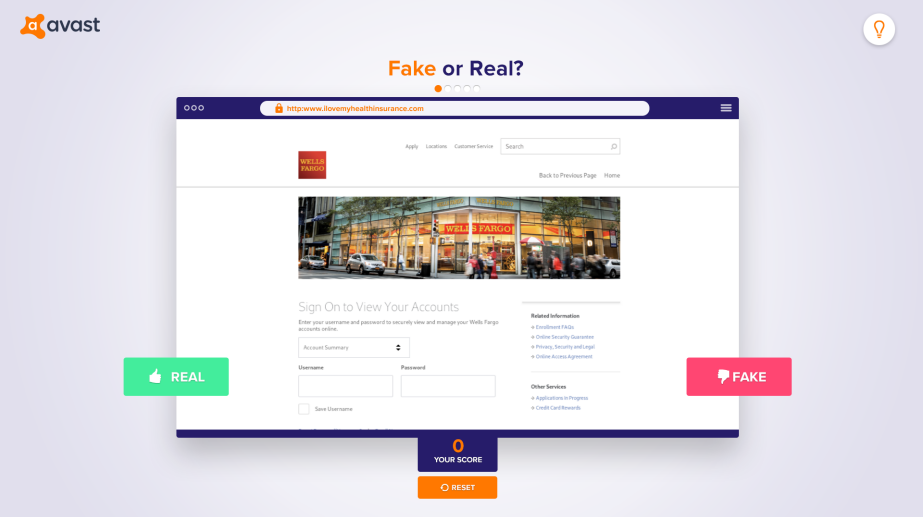


Figure 3.5: Game play screen.

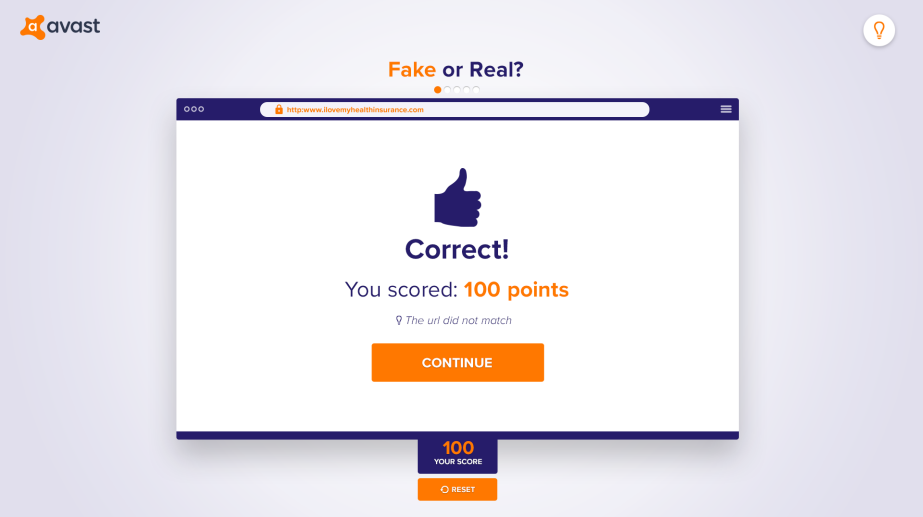


Figure 3.6: Dialogue screen.

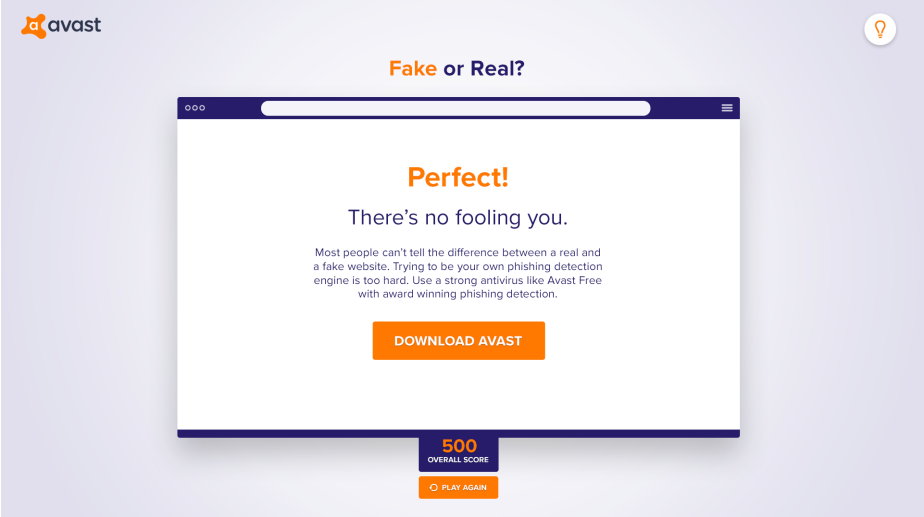


Figure 3.7: Summary screen.

Figure 3.8: Tips screen.

3.2 Build and Deployment

The outcome of this thesis is visible on https://phish.avast.com. The site was implemented using web technologies such as HTML, CSS, JS and NodeJS. The upcoming sections will describe other parts of the implementation which led to the final output.

3.2.1 Data Source

The source data provided by the Avast Threat Lab team consisted of a set of pairs, accessible on a TSV file, with the following format: the image of a real or fake website and the URL. However, we had to add one parameter, which is the *type* of site. The reason for this is because most of the phishing attacks try to replicate well-known websites (e.g. *Amazon*, *Facebook*, *PayPal*). Therefore, the source data contained multiple images of the same legitimate site.

This way, it was simpler to check each record *type* and store it in an array. If the next record would have the same *type*, it would be skipped. This option is optimal, considering that our final set of images is limited to five. If the collection of images would higher, then this implementation would not be efficient due to the high cost of searching. Other data structure could be used instead (e.g. *Binary Search Tree*).

```
1 "015b6cc46e11bec4176cb20429be0b43"              #Fake image
2 "489e387b47fcee4bcf9f506dac6edb52"              #Real image
3 "https://www.paypal-transfertcompte.com/b/"     #Fake URL
4 "https://www.paypal.com/us/signin"              #Real URL
5 "paypal"                                        #Type
```

Code listing 3.1: TSV file containing source data.

The previous solution partially solves the problem of repeated images but does not provide any criteria to generate random content. The result set of items should not have only real or fake sites. For this reason, the *status* parameter is present at the end of each record. This status consists of a random integer between 0 and 1, which represents the real or the fake content of that particular record

```
let con = [
'015b6cc46e11bec4176cb20429be0b43',
'489e387b47fcee4bcf9f506dac6edb52',
'https://www.paypal-transfertcompte.com/b/',
'https://www.paypal.com/us/signin',
'paypal',
'1' ];
let items = [];  /*Max. 5 items*/
items.push(con);
```

Code listing 3.2: Array of items.

This format also allowed to index and retrieve the data instantly to the HTML file.

```
readTsv(function (data) {
    data.forEach((item) => {
        params.items.push({
            'img': '/data/images/' + item[item[5]] + '.jpg',
            'status': item[5],
            'url': item[item[5] + 2]
        })
    })
});
```

Code listing 3.3: Read TSV function in Node JS file.

```
<% items.forEach( item => { %>
<div class="mySlides">
    <span class="img-src" data-status="<%=item.status%>" >
        <img alt="source-img" src="<%=item.img%>">
    </span>
</div>
<% }) %>
```

Code listing 3.4: HTML code calling the read function.

3.2.2 Image Detection

Avast takes into account different factors to recognize phishing sites. Despite the popularity of the domain, suspicious URL or website certificate sometimes is needed to go deeper to determine the legitimacy of a website. Therefore, the Avast Threat Lab team also uses AI together with computer vision methods to analyze images of doubtful sites even further.

Their algorithm uses these methods to choose particular pixels and their surroundings and compare them against potential targets (i.e. images of legitimate sites). Usually, an image containing pixels that are similar to another does not guarantee it belongs to the target set. Hence, spatial verification techniques are used to compare the relation of particular pixels in a picture.

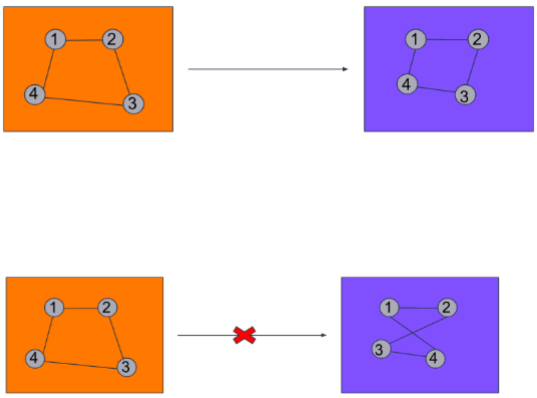


Figure 3.9: Example of accepted and rejected spatial configuration of pixels.

Thanks to Tomáš Trnka (Avast Threat Lab researcher), we could simulate this algorithm. His implementation used *OpenCV*, an open-source computer vision library, to demonstrate in a more graphical way how the spatial verification works. The way the program works is by taking a source (i.e. potential image of a fake website) and a target (i.e. the image of a legitimate website) as input and comparing them. The result is the source image highlighted with the differences against the target.

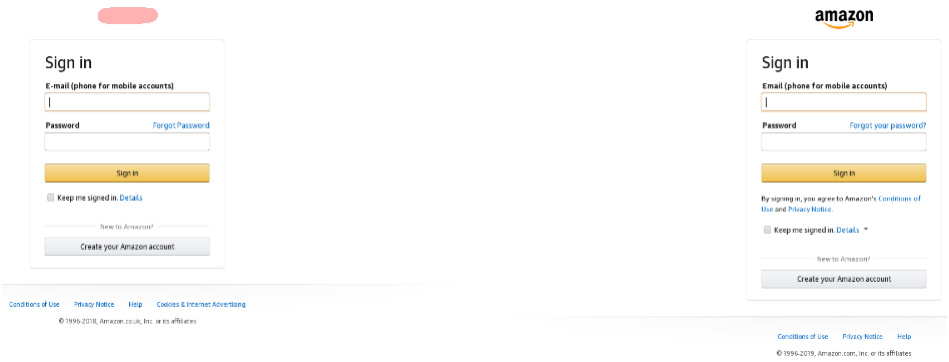


Figure 3.10: Example of *Amazon* output and target image.

This tool allowed to spot some other elements to look out for in suspicious sites, not just the URL, such as missing logos, misleading messages or unusual input fields and buttons. In order to preserve the aesthetics of the application, a different design was used to highlight the differences.

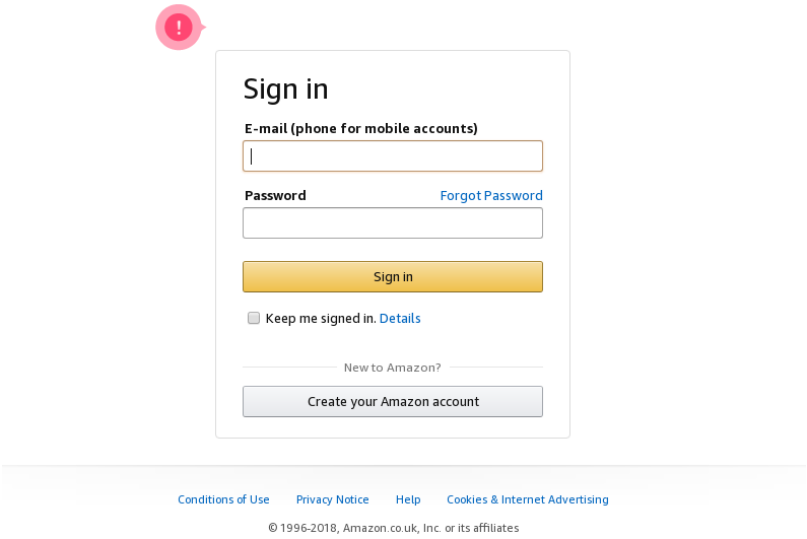


Figure 3.11: Highlight of phishing images.

3.2.3 GA Tracking

For this project, it was not possible to use a database to store game results. Mainly because it would require some testing beforehand, due to security reasons, and postpone the development of the application. However, it is essential to keep track of these results for future research on the topic and make improvements. The solution was to use *Google Analytics* [17] (GA) and *Google Tag Manager* [18] (GTM).

At the end of each game, there is a summary page where users can see their results. There are three types of states based on the score: ***Perfect***, ***Ok*** and ***Disaster***. Using GTM is possible to define tags to label events which are called by triggers and send data to GA. Therefore, each state of the game was label as a ***Results*** event and triggered by data layer from the JavaScript code.

```
1 dataLayer.push({'event': 'Results', 'status': 'Perfect'});
2 dataLayer.push({'event': 'Results', 'status': 'Ok'});
3 dataLayer.push({'event': 'Results', 'status': 'Disaster'});
```

Code listing 3.5: Game events triggered by data layer.

Furthermore, events can also be trigger by HTML elements such as IDs, classes or links. One example of this is the download button, which is triggered by a parameter on the link called ***data-role*** that is used on other Avast websites to track this type of events.

```
<a id="download" class="bi-download-link"
   href="https://www.avast.com/"
   data-role="download-link">
</a>
```

Code listing 3.6: Event triggered by data role.

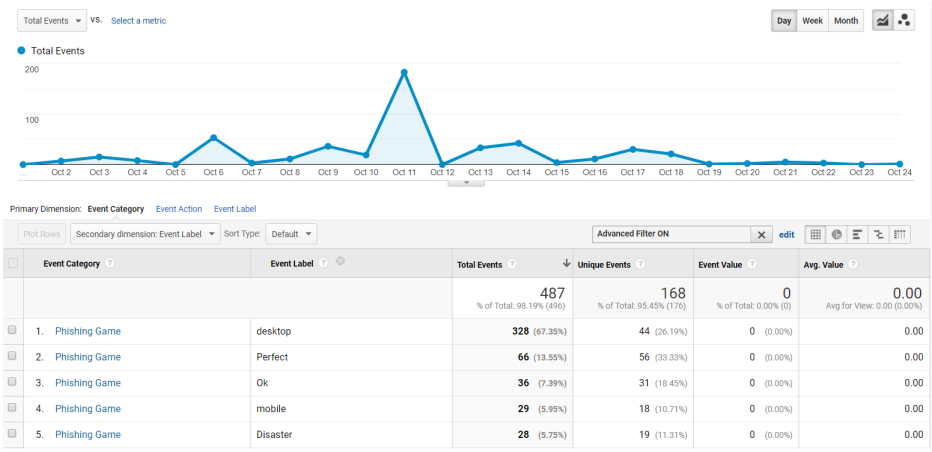


Figure 3.12: Events listed on GA.

3.2.4 Responsiveness

Screen size responsiveness is a must for every website these days. The short time for development did not allow to create a proper mobile design for the game. Therefore, existing elements of the website were modified and adjusted, using CSS media queries, to provide a better mobile experience.

```
@media (max-device-width : 425px)
and (min-device-height : 575px)
and (orientation : portrait) {
    body {
        padding-top : 15%;
    }
}
```

Code listing 3.7: CSS media queries for mobile version.

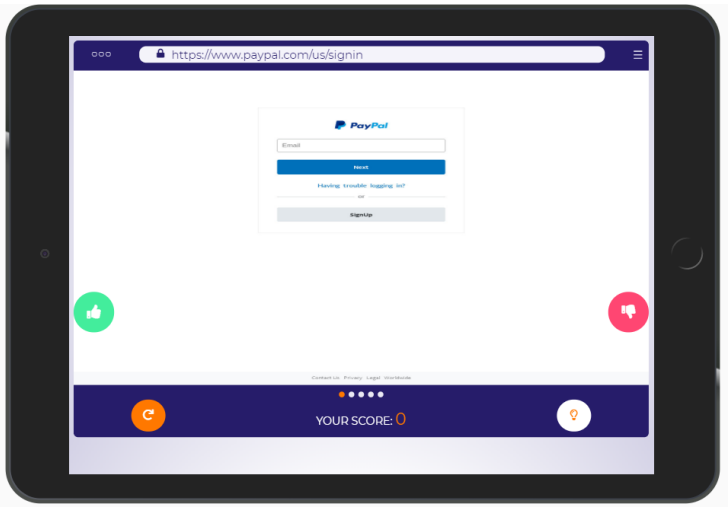


Figure 3.13: Tablet size view of the game.

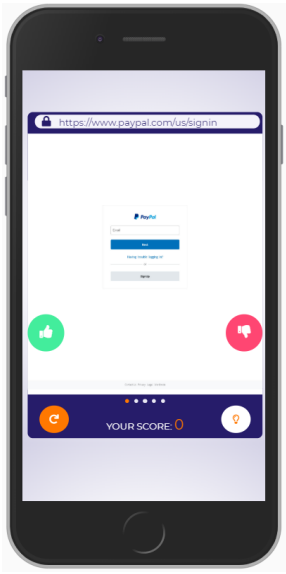


Figure 3.14: Mobile size view of the game.

3.2.5 Server Setup

The last step for deployment was to set up a server where to host the application. The server had to allow to install Node JS packages, install SSL certificates and have accessibility from the client-side. Considering other Avast application are being hosted on the same server, some of the minimum hardware requirements are the following:

- 16GB RAM
- CPU 2 cores
- SSD 128GB

3.3 Testing and Feedback

The first user test, described in section 4.1, was crucial to identify potential issues on the performance of the game. Thanks to these findings, other improvements could be made on time for future events and also were of great value for the overall development. The upcoming sections will describe the main problems in the current implementation and how they were solved.

3.3.1 Issues of the Current Implementation

The main problem with the first version of the game was page loading and response time to user input. It would at least 5 seconds for the website to load and had a slow response on specific elements (e.g. buttons, dialogues). To find the cause of this performance, we used an open-source tool called *Lighthouse* [19] to run an audit on the website. The results showed a low score for performance.

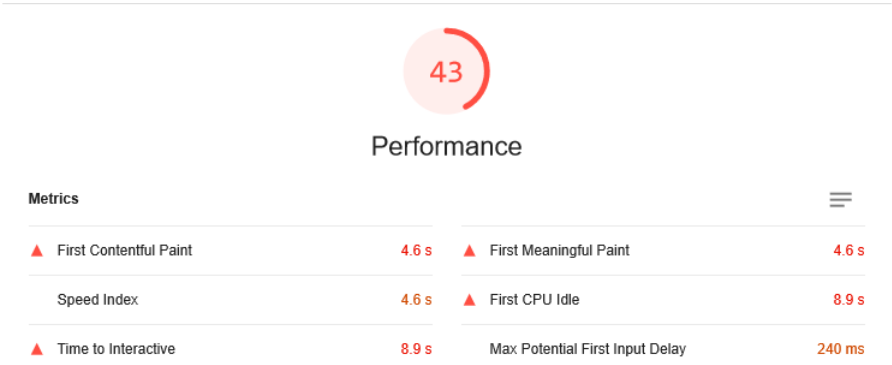


Figure 3.15: Google Audit before improvements.

A significant part of the issue came from the JS code, specifically the use of HTML DOM ***innerHTML*** [20] property. Generally, compared to other DOM manipulation methods (e.g. *JQuery*), innerHTML is most of the times faster to replace HTML code [21]. However, it is not the best practice to use it for raw text since it represents

a security vulnerability. One example of this is the dialogues on the game, which is a generic container that changes continuously by the use of this property.

```
function isPhishing(answer) {
    let dialogueImg = document.getElementById('dialogue-img');
    let dialogueAns = document.getElementById('dialogue-ans');
    if (answer) {
        dialogueImg.innerHTML =
            '<img alt="correct" src="img/icons/correct.png">';
        dialogueAns.innerHTML =
            '<h2>Good job!</h2>';
    }
}
```

Code listing 3.8: Example of *innerHTML* in JS file.

Inserting HTML image tags to the container represented a slowdown on the overall performance.

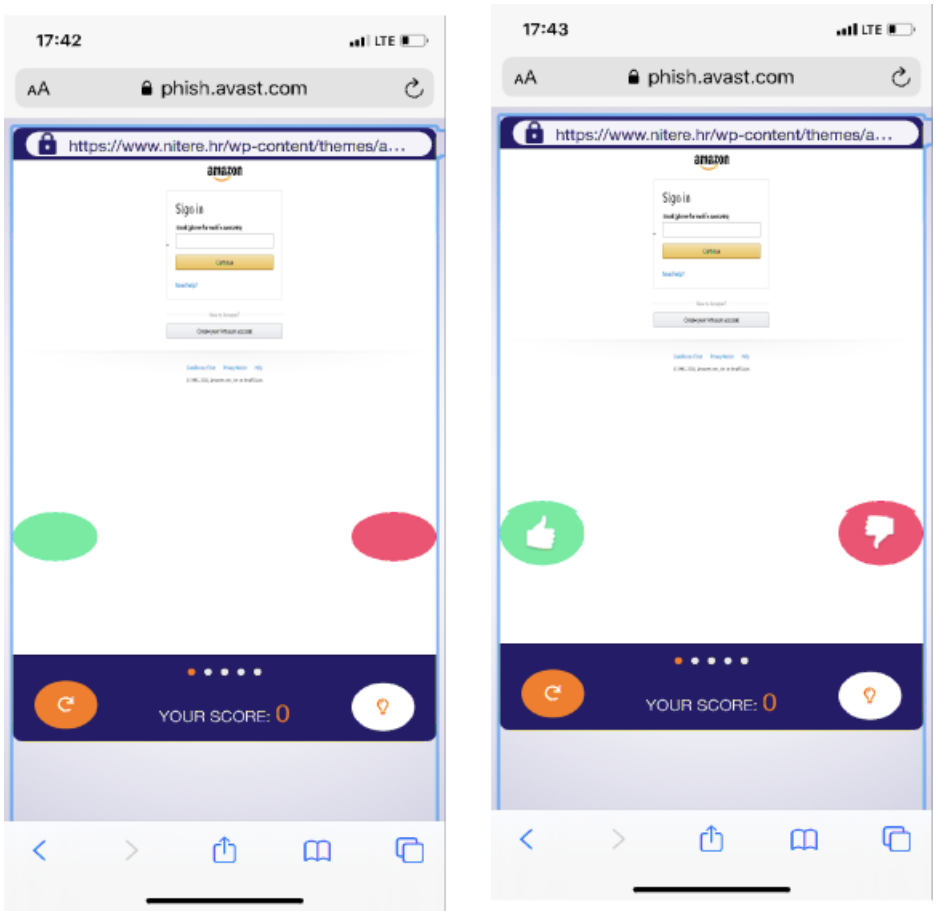


Figure 3.16: Example of delay generated by *innerHTML*.

3.3.2 Improvements

Since icons could easily replace all these image tags, we decided to use an icon toolkit based on CSS called *Font Awesome* [22]. All images inside this container were replaced by icons to keep consistency on the design and avoid user misunderstanding.

```
function isPhishing(answer) {
   let correct = '<i class=\'fas fa-thumbs-up\'
                    style="color: #160E53;
                    font-size: 8em;"></i>';
   if (answer) { dialogueImg.innerHTML = correct; }
}
```

Code listing 3.9: Example of *innerHTML* using icons.

Similarly, removing unnecessary *innerHTML* calls from the JS code. This property parses content as HTML, including spaces, formatting and line breaks. Moreover, if the node includes special characters (e.g. & or <) these are returned as HTML entities (i.e. "&" or "<"). In cases where only raw text needs to be replaced there exist other methods that provide faster performance and do not parse any HTML code (e.g. ***textContent***).

```
function getResults() {
    if (score >= 400) {
        //Before: header.innerHTML = 'Perfect!';
        header.textContent = 'Perfect!';
    }
}
```

Code listing 3.10: Example of updated version using *textContent*.

The use of *textContent* also bypasses security risks. Attackers might use *innerHTML* property to run cross-site scripts. Although, HTML5 would not allow script tags to execute; there are many ways to get away with it. For this reason, it is not recommended to use it to insert plain text [23].

Overall, these changes made a significant impact on the response to user input. Other modifications such as optimizing all other images for web, reducing the number of CSS rules or JS functions helped to improve the page loading.

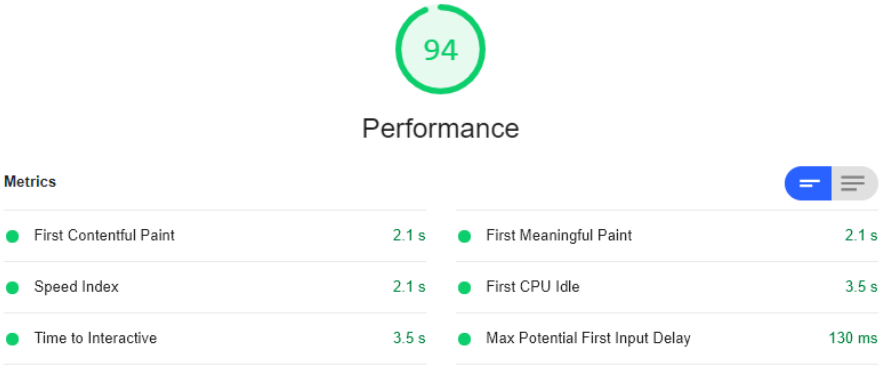


Figure 3.17: Google Audit after improvements.

Chapter 4

Use Cases

The mission of this project is to educate people around the world about phishing and how to identify it. This application will be used mainly on international events related to Cybersecurity or AI, but also on other channels such as social media and blogs. This chapter will go through some real use cases and analyze the data obtained during its use.

4.1 London Science Museum

On July 2019, the London Science Museum opened a new exhibition called *Top Secret* where Avast is one of the main sponsors [24]. Moreover, the museum has themed events the last Wednesday of every month known as *Lates* [25]. Avast hosted the *Lates* event of September, and this represented a unique opportunity to test the game with real users before releasing to the public.

The results on the 25th of September show that 50% of the participant got a ***Perfect*** score. Nevertheless, the other 50% could not spot all the sites correctly, 42% ***Ok*** and 8% ***Disaster*** respectively. This test was helpful to make improvements previously described on section 3.3.2.

Event Category	Event Label	Total Events	Total Events	Contribution to total: Total Events
		12 % of Total: 85.71% (14)	12 % of Total: 85.71% (14)	
1. Phishing Game	Perfect	6	50.00%	
2. Phishing Game	Ok	5	41.67%	
3. Phishing Game	Disaster	1	8.33%	

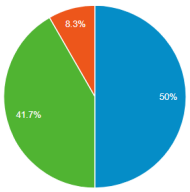


Figure 4.1: *Lates* game results in GA.

4.2 Woman in Business Expo

The Woman in Business Expo [26] is an event that took place in London and gathers together different tech companies to demonstrate the importance of women in the industry. This event was the first time we used the application, after testing and deployment, with attendees that visited the Avast booth. Since we previously set up GTM tags to send data to GA, it was possible to see results from participants of the game.

Event Category	Event Label	Total Events	Total Events	Contribution to total: Total Events
		50 % of Total: 74.63% (67)	50 % of Total: 74.63% (67)	
1. Phishing Game	Perfect	35	70.00%	
2. Phishing Game	Ok	14	28.00%	
3. Phishing Game	Disaster	1	2.00%	

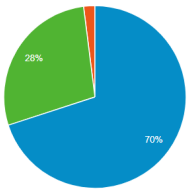


Figure 4.2: Woman in Business Expo game results in GA.

On the results, it is visible that a majority of the participants got a ***Perfect*** score (i.e. 70% of all game events). Even though this looks promising, there is still a 30% that could be potentially a victim of a phishing attack. Another interesting fact is that approximately 20% of all participants clicked on the ***Tips*** button, where they can see some good practices to spot these type of sites.

Event Category	Event Label	Total Events	Total Events	Contribution to total: Total Events
		62 % of Total: 92.54% (67)	62 % of Total: 92.54% (67)	
1. Phishing Game	Perfect	35	56.45%	
2. Phishing Game	Ok	14	22.58%	
3. Phishing Game	mobile	8	12.90%	
4. Phishing Game	desktop	4	6.45%	
5. Phishing Game	Disaster	1	1.61%	

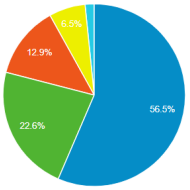


Figure 4.3: Woman in Business Expo events registered in GA for *Tips* button.

During the event, we shared the game on *Twitter* reaching other countries like the United States (47 clicks), Finland (10 clicks) and Germany (3 clicks).

Figure 4.4: *Twitter* post on Avast account.

4.3 Others

After a month of releasing the website to the public and having a reasonable amount of registered game events from other countries, it was also interesting to analyze the results from an international point of view. Considering all users that played the game, the top countries are the Czech Republic with 56%, the United States with 21% and the United Kingdom with 9% of the total user volume. Other countries such as Mexico, Canada or Australia appear on the list.

Regarding game events, the results show that 51% of all users had a ***Perfect*** score, while 28% and 21% received an ***Ok*** and ***Disaster*** score, respectively.

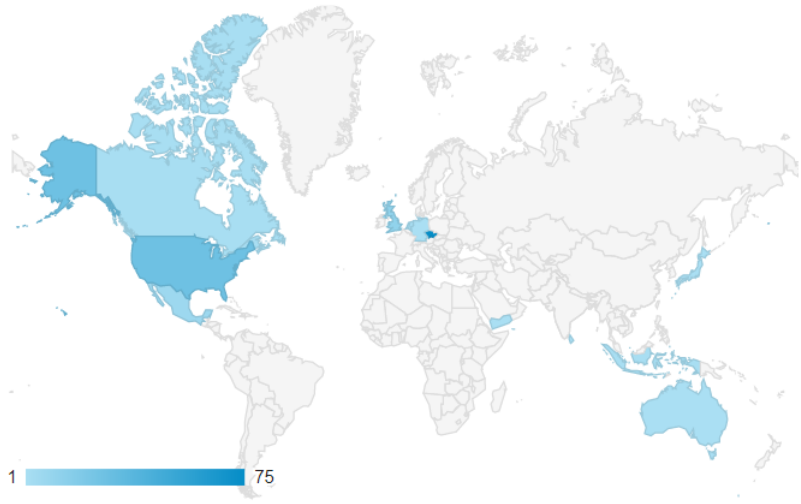


Figure 4.5: Map of user activity in GA.

Event Category	Event Label	Total Events ▼	Total Events	Contribution to total: Total Events ▼
		134 % of Total: 26.75% (501)	134 % of Total: 26.75% (501)	
1. Phishing Game	Perfect	**69**	51.49%	
2. Phishing Game	Ok	**37**	27.61%	
3. Phishing Game	Disaster	**28**	20.90%	

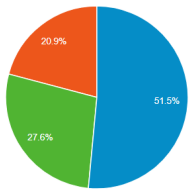


Figure 4.6: Worldwide game results in GA, October 2019.

Chapter 5

Conclusions

In this thesis, we introduced the problem of phishing which is one of the most popular and effective types of cyber-attacks due to its impact and scale that affect millions of users every day. The consequences of successful phishing attacks include loss of sensitive data or even the bankruptcy of businesses. Therefore, the importance to emphasize this matter and educate people about how to recognize and protect from phishing sites.

A gamified web application has been implemented to solve this problem. This application allowed to showcase different types of websites and let users identify whether they are phishing or not. That way, we could make the topic more entertaining and demonstrate some of the most common practices that criminals use.

Data obtained during the use of this application indicated that approximately 30-40 % of people globally were not able to identify the phishing sites presented. Even though these results are not definitive, it correlates with the information researched in the analysis and proves the relevance of the project for the future.

Ideally, the outcome of this thesis will help people to be more engaged and informed about phishing and how to prevent it. Similarly, to keep a positive brand impression of Avast on forthcoming events and fulfil their mission to provide safety around the world.

Further work on this project includes combining the use of AI and computer vision methods presented previously to the game logic. Likewise, to integrate a database that could map the phishing sites stored by Avast directly to the game. Furthermore, extend this project to other regions and languages with local real-time content.

Chapter 6

Acronyms

- ***AI*** - Artificial Intelligence.
- ***CDN*** - Content Delivery Network.
- ***CSS*** - Cascade Style Sheet.
- ***CTA*** - Call To Action.
- ***DOM*** - Document Object Model.
- ***GA*** - Google Analytics.
- ***GTM*** - Google Tag Manager.
- ***HTML*** - Hypertext Markup Language.
- ***HTTPS*** - Hypertext Transfer Protocol Secure.
- ***SaaS*** - Software as a Service.
- ***SSL*** - Secure Socket Layer.
- ***TSV*** - Tab Separated Values.
- ***UI*** - User Interface.
- ***UX*** - User Experience.

Chapter 7

Glossary

- ***Phishing***: Fraudulent attempt to obtain sensitive information such as usernames, passwords and credit card details by disguising oneself as a trustworthy entity in an electronic communication.
- ***Social Engineering***: In the context of information security, is the psychological manipulation of people into performing actions or divulging confidential information.
- ***Smishing***: SMS phishing is a form of criminal activity using social engineering. It uses cell phone text messages to deliver the bait to induce people to divulge their personal information.
- ***Vishing***: Voice phishing is a form of criminal phone fraud, using social engineering over the telephone system to gain access to private personal and financial information for the purpose of financial reward.
- ***Spyware***: Software that aims to gather information about a person or organization, sometimes without their knowledge, and send such information to another entity without the consumer's consent.
- ***Malware***: Any software intentionally designed to cause damage to a computer, server, client, or computer network.
- ***Ransomware***: A type of malware from cryptovirology that threatens to publish the victim's data or perpetually block access to it unless a ransom is paid.

Bibliography

[1] Avast Academy Team. What is phishing, October 2019. [Accessed: 6-October-2019]. Available from: https://www.avast.com/c-phishing.

[2] Avast Academy Team. Social engineering, October 2019. [Accessed: 27-October-2019]. Available from: https://www.avast.com/c-social-engineering.

[3] PhishLabs. Phishing trends and intelligence report. *PhishLabs Annual Report*, pages 13–21, July 2019. [Accessed: 4-November-2019]. Available from: https://info.phishlabs.com/hubfs/2019%20PTI%20Report/2019%20Phishing%20Trends%20and%20Intelligence%20Report.pdf.

[4] Avast Security News Team. Avast collaborates with france and u.s. to stop cryptomining worm, August 2019. [Accessed: 6-October-2019]. Available from: https://blog.avast.com/avast-works-with-france-and-us-to-stop-cryptomining-avast.

[5] Inc PhishMe. Analysis of susceptibility, resiliency and defense against simulated and real phishing attacks. *Enterprise Phishing Resiliency and Defense Report*, page 2, November 2017. [Accessed: 20-October-2019]. Available from: https://cofense.com/wp-content/uploads/2017/11/Enterprise-Phishing-Resiliency-and-Defense-Report-2017.pdf.

[6] Hosting Tribunal. Stunningly scary phishing statistics – an ever-growing threat, June 2019. [Accessed: 21-October-2019]. Available from: https://hostingtribunal.com/blog/phishing-statistics/.

[7] Symantec. Internet security threat report. *ISTR Vol. 23*, page 73, April 2018. [Accessed: 20-October-2019]. Available from: https://www.symantec.com/content/dam/symantec/docs/reports/istr-23-2018-en.pdf.

[8] Alert Logic. Must-know phishing statistics, August 2018. [Accessed: 21-October-2019]. Available from: https://blog.alertlogic.com/must-know-phishing-statistics-2018/.

[9] Debraj Ghosh and Jason Rogers. How office 365 learned to reel in phish, October 2018. [Accessed: 20-October-2019]. Available from: https://www.microsoft.com/security/blog/2018/10/17/how-office-365-learned-to-reel-in-phish/.

[10] APWG. Phishing activity trends report, 4th quarter 2017. *Anti-Phishing Working Group*, page 7, May 2018. [Accessed: 20-October-2019]. Available from: https://docs.apwg.org/reports/apwg_trends_report_q4_2017.pdf.

[11] APWG. Phishing activity trends report, 2nd quarter 2019. *Anti-Phishing Working Group*, page 11, September 2019. [Accessed: 27-October-2019]. Available from: https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf.

[12] Tomáš Trnka. Advances in visual phishing detection. *Avast Blog*, November 2018. [Accessed: 2-November-2019]. Available from: https://blog.avast.com/avast-improves-phishing-detection-avast.

[13] Kevin Townsend. Social engineering – it's not just about phishing. *Avast Blog*, April 2019. [Accessed: 20-October-2019]. Available from: https://blog.avast.com/social-engineering-hacks.

[14] Computerworld. Hackers breach doj, dump details of 9,000 dhs employees, plan to leak 20,000 from fbi, February 2016. [Accessed: 26-October-2019]. Available from: https://www.computerworld.com/article/3030983/.

[15] Avast Security News Team. Ransomware victims pay up and cambridge analytica shuts down, May 2018. [Accessed: 26-October-2019]. Available from: https://blog.avast.com/cambridge-analytica-shuts-down-and-ransomware-victims-pay-up.

[16] Avast Blog. Martin hron, September 2017. [Accessed: 6-October-2019]. Available from: https://blog.avast.com/author/martin-hron.

[17] Google. Google analytics, October 2019. [Accessed: 9-October-2019]. Available from: https://analytics.google.com.

[18] Google. Google tag manager, October 2019. [Accessed: 9-October-2019]. Available from: https://tagmanager.google.com.

[19] Google. Lighthouse, October 2019. [Accessed: 19-October-2019]. Available from: https://developers.google.com/web/tools/lighthouse.

[20] W3Schools. Javascript html dom - changing html, October 2019. [Accessed: 19-October-2019]. Available from: https://www.w3schools.com/js/js_htmldom_html.asp.

[21] Stack Overflow. append vs html vs innerhtml performance, August 2013. [Accessed: 19-October-2019]. Available from: https://stackoverflow.com/questions/18393981/append-vs-html-vs-innerhtml-performance.

[22] Inc Fonticons. Font awesome, October 2019. [Accessed: 9-October-2019]. Available from: https://fontawesome.com/.

[23] Mozilla Developer Network. Element.innerhtml, September 2019. [Accessed: 17-November-2019]. Available from: https://developer.mozilla.org/en-US/docs/Web/API/Element/innerHTML.

[24] London Science Museum. Top secret: From ciphers to cyber security, June 2019. [Accessed: 30-September-2019]. Available from: https://www.sciencemuseum.org.uk/see-and-do/top-secret.

[25] London Science Museum. Lates, September 2019. [Accessed: 30-September-2019]. Available from: https://www.sciencemuseum.org.uk/see-and-do/lates.

[26] Hub Exhibitions Ltd. Women in business expo, September 2019. [Accessed: 20-October-2019]. Available from: https://www.wibexpo.co.uk/.

Appendix A

Contents of enclosed SD Card

- `readme.txt` ..... the file with SD Card content description
- `phishing-game-master` .... the directory of implementation source code
- `thesis` ..... the thesis text directory
  - `BT_Eder_Tejada_2019.pdf` ..... the thesis text in PDF format
  - `src` ..... the directory of LaTeX source codes of the thesis
- `prototypes` ..... the directory of prototypes
  - `wireframes` ..... the directory of wireframes
  - `interface` ..... the directory of interface design
- `reports` ..... the directory of reports
  - `analytics` ..... the directory of GA reports used in PDF format
  - `audit` ..... the directory of Lighthouse audit reports used in PDF format

Implementation source code also available from:
https://github.com/etejada04/Phishing-Game-Final
https://git.int.avast.com/tejadaortigoza/phishing-game